CLAIMS IN CURRENT FORM

(IN FORMAT COMPLIANT WITH THE REVISED 37 CFR 1.121)

1. (ORIGINAL) A method of generating a file suitable for programming a programmable logic device, the method comprising the steps of:

(A) generating a programming item from a plurality of parameters that define a program for said programmable logic device;

(B) compressing said programming item to present a compressed item;

(C) storing said programming item in a programming field of said file in response to generating; and

(D) storing said compressed item in a non-programming field of said file in response to compressing.

2. (ORIGINAL) The method according to claim 1, further comprising the step of storing at least one of said parameters in a second non-programming field of said file.

3. (ORIGINAL) The method according to claim 1, further comprising the step of generating a dictionary for compressing prior to compressing said programming item.

4. (ORIGINAL) The method according to claim 3, wherein said dictionary is generated independently of said compressing step.

5. (ORIGINAL) The method according to claim 4, wherein said compressing is a Huffman encoding and said dictionary is a Huffman tree.

6. (ORIGINAL) The method according to claim 1, further comprising the step of encoding said compressed item from a binary representation to a symbol representation in response to compressing.

7. (ORIGINAL) The method according to claim 6, further comprising the step of mapping said symbol representation to a character representation in response to encoding.

8. (ORIGINAL) The method according to claim 1, further comprising the steps of:

generating an error detection item; and

storing said error detection item in a second non-programming field of said file.

9. (PREVIOUSLY PRESENTED) The method according to claim 8, further comprising the step of:

extracting said error detection item from said file.

10. (ORIGINAL) The method according to claim 1, wherein said steps (A) through (D) are stored in a storage medium as a computer program that is readable and executable by a computer to generate said file.

11. - 20. (CANCELED)

21. (PREVIOUSLY PRESENTED) The method according to claim 1, further comprising the step of adding a plurality of delimiters around said compressed item in said non-programming field.

22. (PREVIOUSLY PRESENTED) The method according to claim 1, wherein said file is compatible with a Joint Electron Device Engineering Council JESD3-C standard.

23. (PREVIOUSLY PRESENTED) The method according to claim 33, further comprising the step of extracting said programming item from said programmable field of said file.

24. (PREVIOUSLY PRESENTED) The method according to claim 23, further comprising the step of replacing said programming item with said backup programming item in response to validating said backup programming item.

25. (PREVIOUSLY PRESENTED) The method according to claim 32, wherein said step of decompressing said compressed item comprises the sub-step of mapping said compressed item from a character representation to a symbol representation in response to extracting said compressed item.

26. (PREVIOUSLY PRESENTED) The method according to claim 25, wherein said step of decompressing said compressed item further comprises the sub-step of decoding said compressed item from said symbol representation to a binary representation in response to mapping.

27. (PREVIOUSLY PRESENTED) The method according to claim 31, wherein said step of extracting said compressed item comprises the sub-step of parsing a plurality of first comments lines containing said compressed item from said file using a plurality of first delimiters.

28. (PREVIOUSLY PRESENTED) The method according to claim 27, wherein said step of extracting said error detection item comprises the sub-step of parsing at least one second comment line containing said error detection item from said file using a plurality of second delimiters.

29. (PREVIOUSLY PRESENTED) The method according to claim 33, further comprising the step of repairing said error detection

item in response to said backup programming item failing said validating step.

30. (PREVIOUSLY PRESENTED) An apparatus comprising:

means for generating a programming item from a plurality of parameters that define a program for a programmable logic device;

means for compressing said programming item to present a compressed item;

means for storing said programming item in a programming field of a file in response to generating; and

means for storing said compressed item in a non-programming field of said file in response to compressing.

31. (PREVIOUSLY PRESENTED) The method according to claim 9, further comprising the step of:

extracting said compressed item from said file.

32. (PREVIOUSLY PRESENTED) The method according to claim 31, further comprising the step of:

decompressing said compressed item to present a backup programming item.

33. (PREVIOUSLY PRESENTED) The method according to claim 32, further comprising the step of:

validating said backup programming item with said error detection item.